

Lámpsakos | N° 11 | pp. 42-49 | enero-junio | 2014 | ISSN: 2145-4086 | Medellín - Colombia

UNA REPRESENTACIÓN LINGÜÍSTICA PARA EVENTOS EN EL CICLO DE VIDA DEL SOFTWARE

A LINGUISTIC REPRESENTATION FOR EVENTS IN SOFTWARE LIFECYCLE

Paola Andrea Noreña-Cardona, Ing.

*Institución Universitaria Tecnológico de Antioquia TdeA
Grupo de Investigación GIISTA
Medellín, Colombia
panorena@tdea.edu.co*

Robinson David González, Estudiante.

*Institución Universitaria Tecnológico de Antioquia - TdeA
Grupo de Investigación GIISTA
Medellín, Colombia
rgonzal4@tdea.edu.co*

Andrés Felipe Mesa-Gómez, Est.

*Institución Universitaria Tecnológico de Antioquia - TdeA
Grupo de Investigación GIISTA
Medellín, Colombia
Amesgo1@tdea.edu.co*

(Recibido el 01-11-2013. Aprobado el 20-12-2013)

Resumen. Un evento es un suceso que ocurre en el transcurso de los procesos de negocio y que afecta los flujos de éstos, generando una causa o un impacto. En el ciclo de vida del software estos elementos carecen de una comprensión lógica para los analistas y desarrolladores, y no se logran traducir correctamente en un proyecto de software. Mediante la definición de reglas lingüísticas se pretende una adecuada representación de los eventos en el ciclo de vida del software. Adicionalmente a partir de las reglas se propone un prototipo que contribuya a la identificación de eventos.

Palabras clave: Artefactos, Eventos, Evento disparador, Evento de resultado, Reglas lingüísticas.

Abstract. An Event is something that happens in the course of business processes and flows affecting these processes, generating a cause or an impact. These elements do not have logical understanding for the analysts and developers, so the events are not translated in a software project correctly. We intend to define linguistic rules for an adequate representation of the events in the software lifecycle. In addition, we propose a prototype to contribute in the identification of events using the linguistic rules defined.

Keywords: Artifacts, Events, Event Trigger, Event Result, Linguistic Rules.

1. INTRODUCCIÓN

Los eventos en el ciclo de vida del software son sucesos que ocurren en un tiempo determinado de los procesos de un sistema y provocan la ocurrencia de cambios en sus estados como la finalización y el inicio de estos procesos [1]. Cuando un evento inicia un proceso se denomina disparador y cuando lo finaliza se denomina de resultado [2 - 3].

En el ciclo de vida se presenta dificultad en la traducción de los elementos de una etapa a otra en un proyecto de software, ya que la comprensión lógica de los requisitos del interesado suele ser diferente en la interpretación de los artefactos que representa el analista y en la implementación que realiza el desarrollador. Algunos de estos elementos que presentan problemas de traducción y representación son los eventos ya que generalmente se confunden con operaciones que realiza un usuario en la interfaz gráfica GUI, como dar un clic o seleccionar una opción de un menú [4].

De acuerdo con lo anterior, en este artículo se pretende definir reglas lingüísticas para la determinación de eventos, que mejoren su traducción y representación en el ciclo de vida del software. Para observar la funcionalidad de las reglas, se propone un prototipo que contribuya a la determinación de eventos.

Este artículo tiene la siguiente estructura: en la sección 2 se definen los eventos; En la sección 3 se plantea el problema de representación de eventos; en la sección 4 se definen algunas reglas lingüísticas; en la sección 5 se muestra cómo el prototipo aplica las reglas. Finalmente se presentan las conclusiones y el trabajo futuro.

2. EVENTOS

Es un suceso que pasa en el transcurso de los procesos de negocio y que afecta los flujos de estos procesos, generando una causa o un impacto [5 - 6]. Un evento es la especificación de un acontecimiento significativo, ubicado en el tiempo y en el espacio [6]. El término “evento” es lo suficientemente general como para abarcar los diferentes instantes en un proceso de negocio: el inicio de una actividad, el final de una actividad, el cambio de estado de un proceso, un mensaje que llega, entre otros [2].

Todos los flujos deben tener un inicio que se da por un evento disparador. El orden de ocurrencia de los procesos se establece a partir de la circunstancia disparadora, indicando su punto de partida. Estos eventos sólo tienen flujos de secuencia de salida y pueden desencadenar, también, la ejecución de otros eventos. Un inicio de estos flujos puede deberse a: una declaración, una restricción o una acción. Cada disparador puede tener varias acciones asociadas [7 - 8].

3. DESCRIPCIÓN DEL PROBLEMA

Los eventos en el ciclo de vida del software, son elementos necesarios en los flujos de los procesos de negocios. Estos presentan la dificultad de interpretarse como aquellas operaciones que realiza un usuario en la interfaz gráfica GUI, como dar un clic o seleccionar una opción de un menú [4]. Se requiere para algunos artefactos que representan eventos como el diagrama de procesos de negocios (BPMN), el diagrama de máquina de estados y el diagrama de actividades, que los eventos a partir de los requisitos del interesado permitan la traducción en la etapa de implementación en una solución del software, ya que los eventos cambian los estados de los procesos y afecta el flujo de la secuencia de las actividades [10].

Es importante una correcta traducción e interpretación de los eventos ya que estos controlan el inicio y el fin de los flujos de los procesos, por tanto se deben representar para observar los procesos del negocio e imprimirlos de la manera adecuada.

4. ALGUNAS REGLAS LINGÜÍSTICAS PARA LA DETERMINACIÓN DE EVENTOS EN EL CICLO DE VIDA

Se pretende dar solución a la correcta definición de los eventos a partir de reglas lingüísticas que permitan mejorar su traducción e interpretación en el ciclo de vida del software, teniendo en cuenta algunas características lingüísticas que intentan modelar de forma lógica el lenguaje natural desde un punto de vista computacional [11].

Partiendo de los verbos como una característica lingüística principal en oraciones [12] y por ende dentro del contexto de las necesidades de los interesados, es posible generar expresiones dentro

Tabla 1. Reglas eventos disparadores



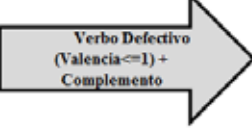

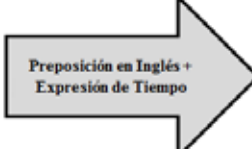

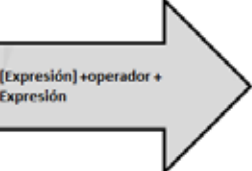

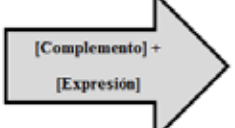

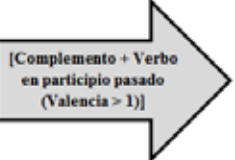

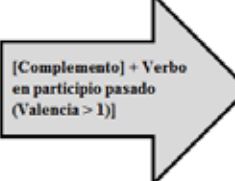
TIPO EVENTO	DEFINICIÓN	REGLAS	EJEMPLOS
MENSAJE O SEÑAL	<p> VERBO PRESENTE: se utilizan sin sujeto, solo se conjugan en tercera persona.</p> <p>VALENCIA: Entre 0 y 1</p> <p> COMPLEMENTO: Tema o sujeto impersonal.</p>	<p> Verbo Defectivo (Valencia <= 1) + Complemento</p>	<p><i>Aparece orden, Llega pedido, Comienza tarea.</i></p>
TIEMPO	<p> PREPOSICIÓN EN INGLÉS: Absoluta: <i>At</i> (<valor temporal>), Relativa: <i>After</i> (<valor temporal>)</p> <p>EXPRESIÓN DE TIEMPO: Cantidad de Tiempo determinado, Hora de Inicio, Fecha, Rango de tiempo, día, puede colocarse como una regla.</p>	<p> Preposición en Inglés + Expresión de Tiempo</p>	<p><i>At (12:00pm), At (12/12/2012), after (5:00pm), At (Friday)</i></p>
CONDICIONAL	<p> INSTRUCCIÓN CONDICIONAL:</p>	<p> [Expresión] + operador + Expresión</p>	<p><i>Saldo > 0, Tipo de Pago / (Credito, Efectivo)</i></p>

Tabla 2. Reglas eventos de resultado

TIPO EVENTO	DEFINICIONES	REGLA	EJEMPLOS
EXCEPCIÓN	<p> COMPLEMENTO: Tema o sujeto sobre el cual se genera el error.</p> <p>EXPRESIÓN: Expresa que error podría darse en un flujo.</p>	<p> [Complemento] + [Expresión]</p>	<p>Login (<i>Usuario Incorrecto</i>)</p>
CANCELADO	<p> VERBO: Forma no personal del verbo, participio pasado denotando que la acción se realizó.</p> <p>VALENCIA: Superior a 1</p> <p>COMPLEMENTO: Tema o sujeto por el cual la acción se cancela o finaliza</p>	<p> [Complemento + Verbo en participio pasado (Valencia > 1)]</p>	<p>Conexión <i>Terminada</i>, Venta <i>Cancelada</i></p>
TERMINADO	<p> VERBO: Forma no personal del verbo, porque normalmente no se utiliza una persona para recaer la acción, utilizan un sujeto, su uso sera en participio pasado denotando que la acción se realizó.</p> <p>VALENCIA: Superior a 1</p> <p>COMPLEMENTO: Tema o sujeto para el cual la acción término, o finalizo</p>	<p> [Complemento] + Verbo en participio pasado (Valencia > 1)</p>	<p>Orden <i>Despachada</i></p>

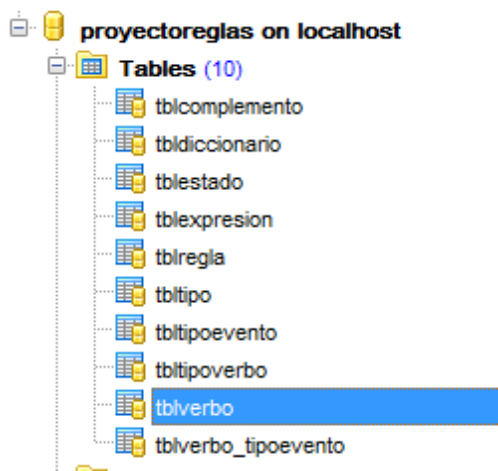


Figura 1. Base de datos

de reglas lingüísticas que permitan interpretar y traducir eventos desde el proceso de educación de requisitos hasta la implementación de una solución de software.

Las valencias especifican semánticamente las frases en el contexto del verbo. El papel que desempeñan las valencias tiene una correspondencia directa con la situación lexicográfica y se requieren para codificar la información concerniente al contexto y al orden de las palabras, con el fin de limitar el análisis y la generación del lenguaje natural. Existen verbos sin valencia, con una valencia y con más de una valencia. Los verbos sin y con una valencia son aquellos que se conjugan en tercera persona como: llover, granizar, nevar y que no requieren hablar de un sujeto determinado; cuando esto sucede se habla de eventos ya que no tienen una persona que lo ejecute [13].

En la tabla 1 se presentan las reglas para ciertos eventos disparadores y en la tabla 2 se muestran los resultados a partir de la representación del BPMN, donde se define de acuerdo con la valencia cómo puede representarse lingüísticamente de forma correcta. También se presenta definiciones y ejemplos, que permiten realizar una mejor obtención de requisitos, lo cual y facilita a los analistas y desarrolladores la interpretación y traducción de los eventos.



Figura 2. Pantalla principal

5. PROTOTIPO

En las funciones del prototipo de reglas lingüísticas se encuentran: la identificación de eventos en la etapa de análisis y la definición de verbos según el tipo de valencias que se pueden aplicar en las reglas a los eventos, con el objetivo de que los analistas puedan definirlos adecuadamente en los artefactos que se representan.

5.1. COMPONENTES

Base de Datos: Se creó una base de datos para almacenar las reglas utilizando el motor MySQL 5.3 gratuito, y las diferentes características lingüísticas como: los verbos con los cuales, se hizo una gran labor para tratar de ingresar la mayor parte de verbos con sus respectivas valencias, las preposiciones, los artículos y complementos como se puede observar en la figura 1.

Servicio Web JAVA: Contiene toda la lógica del prototipo; se desarrolló en un servicio web para facilitar la integración con cualquier otro sistema realizado que debe ser desplegado en servidores; en este prototipo se trabajó con Glassfish 4.0 servidor gratuito.

Cliente VB.Net: Aplicación de Escritorio que consume el servicio web y trabaja la información para mostrarla al usuario en distintos formularios de una manera más tangible y organizada.

5.2 FUNCIONALIDAD PROTOTIPO

Como se puede observar en la figura 2, Desde una pantalla principal se accede a las diferentes opciones del sistema desde el menú operaciones,



Figura 3. Búsqueda de Eventos

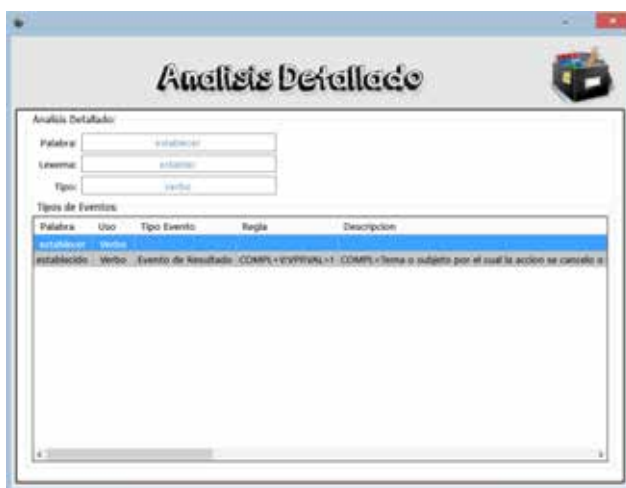


Figura 4. Análisis Detallado



Figura 5. Definición de Eventos

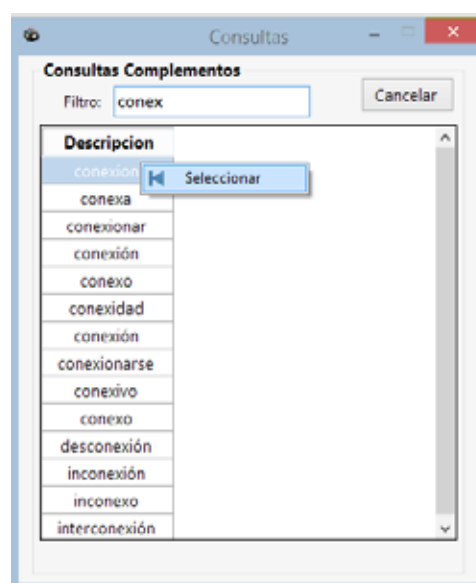


Figura 6. Consultas complementos

La primera opción dirige hacia el formulario de búsqueda de eventos (ver figura 3), mediante la primera parte en un párrafo escrito por el interesado en la fase de requisitos; para ello se ingresa un texto no mayor a 500 caracteres extraído del artefacto, y se presiona el botón buscar. Para el ejemplo se presenta en el texto: “Al establecer la conexión se ejecutará la consulta”. En la segunda parte se presenta el análisis de las palabras operando como un diccionario léxico, el sistema separa las palabras y de acuerdo con su lexema identifica en colores las partes del texto: si es

verde identifica un verbo, si es rojo, una expresión (artículo o preposición) y si es azul, un complemento. Estos resultados se presentan desde la base de conocimiento, realizando varias validaciones para encontrar la posible existencia de eventos, que se logra mediante la búsqueda del número de valencias que contiene el verbo.

Para observar el análisis detallado (función principal del sistema) de cada palabra, se da clic sobre cada una.



Figura 7. Resultado de búsqueda complemento



Figura 9. Definición de Eventos

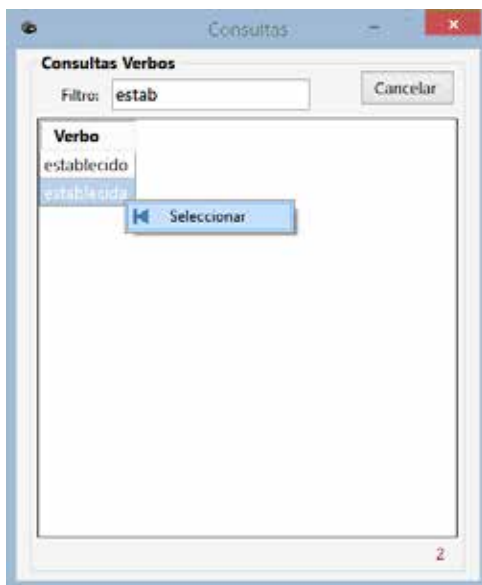


Figura 8. Consultas de verbos

Para observar la funcionalidad del análisis detallado se seleccionó la palabra *establecer*, como se observa en la figura 4, se encuentra en verde indicando que es un verbo. En esta nueva pantalla muestra la palabra, su lexema y su clasificación. En la segunda parte muestra la aplicación de la regla que restringe el verbo establecer como un evento de resultado, según la regla en participio pasado (ver tabla 2), determinando sus posibles usos en un evento de tipo resultado, el cual se daría según la regla que muestra en pantalla $COMPL+V:VPP.VAL>1$. Cada variable se explica en la columna descripción $COMPL=$ Tema o sujeto por el cual la acción se canceló o finalizó

inoportunamente. | $VPP=$ Forma no personal del verbo, porque normalmente no se utiliza una persona para recaer la acción, utilizan un sujeto, su uso será en participio pasado denotando que la acción se realizó. $VAL=$ Superior a 1 | EJEMPLO= Conexión Terminada, Venta Cancelada, así en el texto se encuentra un evento en el proceso, y presenta la manera correcta de representarse lingüísticamente, en este caso define *establecido* como un evento de resultado y puede indicar que algo fue terminado; así debería ser indicado en todos los artefactos que se represente. Esto es un elemento diferenciador de los demás diccionarios léxicos ya que identifican según las valencias la aplicación de las reglas para eventos en el ciclo de vida del software; de esta manera es una herramienta de apoyo para los analistas y permite dar claridad de estos importantes elementos en los artefactos que modelan procesos que luego serán implementados.

La segunda opción que tiene el menú principal es la definición de eventos. Este servicio es una guía para la correcta definición de eventos en caso de tener el evento desconociendo la manera de definirlo. En este caso se elige el tipo de evento que se desea definir y la opción de aceptar. El sistema presenta la regla y muestra las opciones de búsqueda basado en las partes de la regla. Así en la figura 5 se muestra como ejemplo la elección de un evento de resultado donde se presenta su respectiva regla, de acuerdo con esto, trae sus dos partes, complemento y verbo y la opción de buscarlas ya que se establece con base en sus valencias.

Al ingresar la palabra, busca por su lexema del complemento que acompañará el verbo; en el ejemplo busca el lexema *conex* para el complemento *conexión* (ver figura 6). Se inicia la búsqueda llevando a una nueva ventana de consultas de complemento que muestra todas las opciones para la palabra, luego se selecciona con clic derecho y clic en Ítem Seleccionar. Automáticamente se cargará en la caja de texto permitiendo la opción de búsqueda para el verbo como se muestra en la figura 7.

Para el complemento *conexión* existe un verbo que se asocia a una valencia, por tanto, al buscarlo le dará exclusivamente las opciones correctas de definición del verbo según la regla del tipo de evento. Al igual que en los complementos, se escribe el lexema, en este caso *estab* y se da clic en la opción buscar; el sistema filtra y despliega una nueva ventana de consulta de verbos, donde presenta las opciones más apropiadas para que el analista elija la correspondiente. En el ejemplo se evidencian dos opciones *establecido* y *establecida*; de acuerdo con la palabra *conexión* se debe seleccionar *establecida* para acompañar la forma femenina de conexión como se puede ver en la figura 8.

Para finalizar el proceso se presenta al final la definición correcta del evento según la regla lingüística como se puede observar en la figura 9.

6. CONCLUSIONES Y TRABAJO FUTURO

En el ciclo de vida del software intervienen elementos importantes como los eventos que inciden radicalmente en los procesos, ya que informan acerca de los sucesos importantes dentro del sistema.

En un esfuerzo por mejorar el proceso en el ciclo de vida del software se logran identificar algunas necesidades en la representación de elementos en los procesos de negocio como los eventos, así se realizaron algunas reglas lingüísticas y un prototipo de definición de eventos como herramientas de apoyo en interpretación y traducción para analistas y desarrolladores en artefactos que representan eventos.

Como posibles trabajos futuros se puede dar en la herramienta una definición más detallada de los tipos de eventos de resultado y los tipos de eventos disparadores que se encuentran en el texto a partir de reglas precisas lingüísticamente.

AGRADECIMIENTOS

Se agradece a Colciencias, Departamento Administrativo de Ciencia, Tecnología e Innovación en Colombia por su aporte y financiación al proyecto de investigación “Un modelo de representación para los eventos en el ciclo de vida del software” en el que se enmarca este artículo, en la convocatoria de jóvenes investigadores.

REFERENCIAS

- [1] N. Weinbach, A. García. “Una Extensión de la Programación en Lógica que incluye Eventos y Comunicación”. VI Workshop de Investigadores en Ciencias de la Computación, Buenos Aires (Argentina) pp 446-451, Mayo 2004.
- [2] Oracle. “Documento número: B14220-02 Documento estándar” Disponible en: http://docs.oracle.com/cd/B19306_01/server.102/b14220/triggers.htm , 2005. Accedido en Mayo, 2014.
- [3] F. Arango, C. M. Zapata. “UN-MÉTODO para la Elicitación de Requisitos de Software”. Tesis Doctoral en Ingeniería. Universidad Nacional de Colombia, Medellín, 2006.
- [4] F. Osorio. “Lógica y programación orientada a los objetos: un inicio al desarrollo de software.” Tesis de Ingeniería, Instituto Tecnológico Metropolitano Medellín, Colombia. 2008.
- [5] OMG (Object Management Group) “Business Process Model and Notation BPMN”. Standard Document. 2009. Disponible en: <http://www.omg.org/spec/BPMN/1.2>
- [6] M. Owen & O. Raj. “BPMN and Business Process management Introduction to the New Business Process Modeling Standard”. Popkin Software, 2003.
- [8] M. Fowler. “UML distilled a brief guide to the standard object modeling language”. 2° Ed. Pearson Education, Boston, MA (USA). 2000.
- [9] S. A. White, M. Derek. “Guía de Referencia y Modelado BPMN”. Future Strategies Incorporated, 218p, 2010.

- [10] U. Hernández; F.J. Álvarez and M. Vargas. "Use Processes - Modeling Requirements Based on Elements of BPMN and UML Use Case Diagrams". *Second International Conference on Software Technology and Engineering (ICSTE)*. 2010.
- [11] A. Igor, L. Bolshakov, A. Gelbukh. "IPN-UNAM-FCE, Computational Linguistics". Computational Linguistics: Models, Resources, Applications. IPN-UNAM-FCE, 186 p, 2004.
- [12] V. Llamazares. "Aproximación a la lingüística computacional". Tesis Doctoral en Ingeniería, Universidad de León, España. 2010.
- [13] S. Galicia, A. Gelbukh, "Investigaciones en análisis sintáctico para el español." México DF, 291 México: Conacyt. 2007
- [14] U. Hernández; F. J. Álvarez and M. Vargas. "Use Processes - Modeling Requirements Based on Elements of BPMN and UML Use Case Diagrams". *2nd International Conference on Software Technology and Engineering (ICSTE)*. 2010.